IN THE CLAIMS

Claim 1 (Currently Amended): A process for providing a representation of specified characteristics of a previously developed object-oriented software program, ~~said~~ the software program including a ~~number~~ plurality of object classes and further including object related methods belonging to respective object classes, ~~said~~ the process comprising ~~the steps of~~:

sensing at least one complex method call ~~included in said~~ by examining the source code of the software program, a plurality of ~~said~~ the methods being associated with each of ~~said~~ the at least one complex method ~~calls~~ call, wherein the at least one complex method call includes at least one single method call;

extracting ~~a number of~~ the at least one single method ~~calls~~ call from each of ~~said~~ the at least one complex method ~~calls~~ call, the extracting comprises recursively replacing the at least one single method call with a phase variable;

generating a set of information for each of ~~said~~ the methods from ~~said~~ the at least one single method ~~calls~~ call, the information set for a particular method containing at least ~~the~~ a name of the particular method and the object class to which the particular method belongs; and

constructing a representation of interactions between objects of ~~said~~ the software program from the information contained in ~~said~~ the method information sets.

Claim 2 (Currently Amended): The process of ~~Claim~~ claim 1, wherein[[:]] ~~said extracting step comprises replacing a component of a complex method call with a phase variable to produce a method call of reduced complexity~~ the at least one single method call comprises a method parameter of the at least one method call or a continuous method call.

Claim 3 (Currently Amended): The process of ~~Claim 2~~ claim 1, wherein[[:]] ~~said~~ the process further comprises ~~includes an initial step of~~ extracting the name and class of each of ~~said~~ the methods from ~~said~~ the software program.

Claim 4 (Currently Amended): The process of ~~Claim~~ claim 1, wherein[[:]] ~~a given complex method call comprises multiple method related components, and said~~ the extracting ~~step~~

comprises ~~recursively substituting a phase variable for each of said method related components, until said~~ output a given complex method call ~~has been resolved into~~ as multiple lines, ~~each containing one of said~~ wherein at least one of the multiple lines contains the at least one single method ~~calls~~ call.

Claim 5 (Currently Amended): The process of ~~Claim~~ claim 4, wherein ~~said~~ the extracting ~~step~~ comprises:

a first parsing phase ~~disposed~~ configured to separate any casting operations included in ~~said~~ the given complex method call;

a second parsing phase ~~disposed~~ configured to isolate any method parameters included in ~~said~~ the given complex method call; and

a third parsing phase ~~disposed~~ configured to resolve any continuous method calls included in ~~said~~ the given complex method call into multiple lines, each containing one of ~~said~~ the at least one single method ~~calls~~ call.

Claim 6 (Currently Amended): The process of ~~Claim~~ claim 5, wherein[[:]] ~~said~~ the first parsing phase is implemented prior to ~~said~~ the second parsing phase, and ~~said~~ the second parsing phase is implemented prior to ~~said~~ the third parsing phase.

Claim 7 (Currently Amended): The process of ~~Claim~~ claim 6, wherein[[:]] ~~said~~ the ~~step of~~ generating ~~method information sets~~ includes parsing an output provided by ~~said~~ the third parsing phase to determine the correct object class for each of ~~said object related~~ the methods.

Claim 8 (Currently Amended): The process of ~~Claim~~ claim 7, wherein[[:]] ~~said~~ the process further comprises ~~includes the step of~~ determining whether a method is a user-defined method or a standard ~~API~~ application programming interface method.

Claim 9 (Currently Amended): The process of ~~Claim~~ claim 1, wherein[[:]] ~~said~~ the constructing ~~step~~ comprises constructing a sequence diagram depicting the interactions between respective objects of ~~said~~ the software program.

Claim 10 (Currently Amended): The process of ~~Claim 1~~ claim 9, wherein[[:]] the sequence diagram displays ~~the~~ a condition of a method call to indicate that the method call occurs only when the condition is evaluated to be true.

Claim 11 (Currently Amended): The process of ~~Claim~~ claim 1, wherein[[:]] ~~said~~ the software program is in the form of source code.

Claim 12 (Currently Amended): The process of ~~Claim1~~ claim 1, wherein[[:]] ~~said~~ the software program is written in ~~Java~~ JAVA™ software code.

Claim 13 (Currently Amended): The process of ~~Claim~~ claim 1, wherein[[:]] ~~said~~ the software program is written in C++ software code.

Claim 14 (Currently Amended): The process of ~~Claim~~ claim 1, wherein[[:]] at least one of ~~said~~ the object related methods in ~~said~~ the software program is a polymorphic method.

Claim 15 (Currently Amended): The process of ~~Claim~~ claim 1, wherein[[:]] at least one of ~~said~~ the object related methods in ~~said~~ the software program is related to an inheritance feature, and ~~said~~ the extraction ~~step~~ includes tracking an inheritance path until it reaches a parent object class ~~wherein~~ to which the method is defined.

Claim 16 (Currently Amended): A computer program product in a computer readable media ~~system~~ for providing a representation of specified characteristics of a previously developed object-oriented software program, ~~said~~ the software program including a ~~number~~ plurality of object classes and object related single methods belonging to respective object classes, ~~said~~ the software program further including at least one complex method call containing a plurality of ~~said~~ the single methods, ~~said system~~ the computer program product comprising:

~~a Method Detail Parser unit disposed to extract a number~~ first instructions for extracting a plurality of individual method calls from each of ~~said~~ the at least one complex method ~~calls~~ call by examining the source code of the software program, wherein the first instructions comprise recursively replacing each of the single methods with a phase variable;

second instructions for storing in a data base ~~operable to store~~ a set of information for each of ~~said~~ the single methods, the set of information ~~set~~ for a particular single method containing at least ~~the~~ a name of the particular method and the object class to which the particular method belongs; and

third instructions for graphically constructing and graphically outputting ~~a drawing device operable to construct~~ a representation of interactions between objects of ~~said~~ the software program from the information contained in ~~said method~~ the set of information ~~sets~~.

Claim 17 (Currently Amended): The ~~system~~ computer program product of ~~Claim~~ claim 16, wherein[[:]] ~~said system includes a Method Information Parser unit disposed to extract~~ further comprising fourth instructions for extracting the name and the object class of each of ~~said~~ the single methods from ~~said~~ the software program.

Claim 18 (Currently Amended): The ~~system~~ computer program product of ~~Claim~~ claim 17, wherein[[:]] ~~said Method Detail Parser unit is disposed to recursively substitute~~ the first instructions comprise fifth instructions for outputting ~~a phase variable for each of a plurality of method related components contained in~~ a given complex method call~~, until said given complex method call has been resolved into~~ as multiple lines, each containing one of ~~said~~ the single method calls.

Claim 19 (Currently Amended): The ~~system~~ computer program product of ~~Claim~~ claim 18, wherein[[:]] ~~said the Method Detail Parser unit is~~ the first instructions sequentially ~~operated to~~ implement a first parsing phase to separate any casting operations included in ~~said~~ the given complex method call, ~~to implement~~ a second parsing phase to isolate any method parameters included in ~~said~~ the given complex method call, and ~~to implement~~ a third parsing phase to resolve ~~said~~ the given complex method call into multiple lines, each containing one of ~~said~~ the single method calls.

Claim 20 (Currently Amended): The ~~system~~ computer program product of ~~Claim~~ claim 19, wherein[[:]] ~~said drawing device is operable to construct~~ the third instructions

constructs a sequence diagram depicting the interactions between respective objects of ~~said~~ the software program.

Claim 21 (Currently Amended): The ~~system~~ computer program product of ~~Claim~~ claim 20, wherein[[:]] ~~said~~ the software program is in the form of source code.

Claim 22 (Currently Amended): An apparatus ~~Apparatus~~ for providing a sequence diagram representing specified characteristics of a previously developed object-oriented software program, ~~said~~ the software program including a ~~number~~ plurality of object classes and further including object related methods belonging to respective object classes, ~~said~~ the apparatus comprising:

a computer system having a display;

means for sensing at least one complex method call ~~included in said~~ by examining the source code of the software program, a plurality of ~~said~~ the methods being associated with each of ~~said~~ the at least one complex method ~~calls~~ call;

Method Detail Parser means for extracting a ~~number~~ plurality of single method calls from each of ~~said~~ the at least one complex method ~~calls~~ call, the Method Detail Parser means comprises recursively replacing the associated methods with a phase variable;

means for generating a set of information for each of ~~said~~ the object related methods from ~~said~~ the single method calls, the set of information ~~set~~ for a particular object related method containing at least ~~the~~ a name of the particular method and the object class to which the particular method belongs; and

means for constructing a sequence diagram representing interactions between objects of ~~said~~ the software program from the information contained in ~~said method~~ the sets of information ~~sets~~ and outputting the sequence diagram on the display.

Claim 23 (Currently Amended): The apparatus ~~if Claim~~ of claim 22, wherein[[:]] ~~said~~ the apparatus ~~includes~~ further comprises Method Information Parser means for extracting the name and the object class of each of ~~said~~ the methods from ~~said~~ the software program.

Claim 24 (Currently Amended): The apparatus of ~~Claim~~ claim 23, wherein[[:]] ~~said~~ the Method Detail Parser means is operable to ~~recursively substitute a phase variable for each of a plurality of method related components contained in~~ output a given complex method call~~, until said given complex method call has been resolved into~~ as multiple lines, each containing one of ~~said~~ the single method calls.

Claim 25 (Currently Amended): The apparatus of ~~Claim~~ claim 24, wherein[[:]] ~~said~~ the Method Detail Parser means is ~~disposed~~ configured to separate any casting operations included in ~~said~~ the given complex method call during a first parsing phase, to isolate any method parameters included therein during a second parsing phase, and resolve ~~said~~ the given complex method call into multiple lines, each containing one of ~~said~~ the single method calls, during a third parsing phase.

Claim 26 (Currently Amended): The apparatus of ~~Claim~~ claim 25, wherein[[:]] ~~said~~ the first parsing phase is implemented prior to ~~said~~ the second parsing phase, and ~~said~~ the second parsing phase is implemented prior to ~~said~~ the third parsing phase.

Claim 27 (Currently Amended): The apparatus of ~~Claim~~ claim 26, wherein[[:]] ~~said~~ the constructing means comprises a drawing engine for depicting interactions between respective objects of ~~said~~ the software program.

Claim 28 (Currently Amended): The apparatus of ~~Claim~~ claim 27, wherein[[:]] ~~said~~ the software program is in the form of source code.